Brief Introduction to Machine Learning and Big Data Project

Ritai Na

1. Introduction:

Shown below is a brief Introduction to Wind Turbine Status Analysis Project in the course Machine Learning and Big Data:

You may check the code in Github link Below.

Github Project Link: https://github.com/BlueFamous/Machine-Learning-Course-Project

2. Data Preprocessing:

(1) Dimension Reduction: The original dataset consist of over 67 attributes and the total lines of data is over 1 million. To increase efficiency, I applied Lasso to make dimension regression using R.

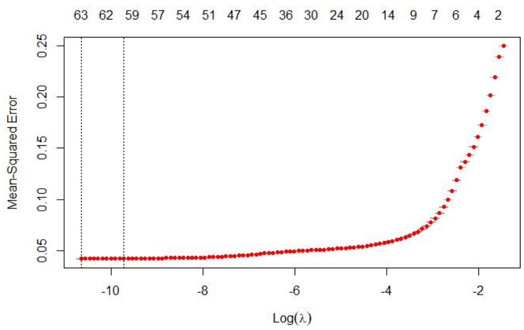

Fig.1 Result of the Lasso Regression

(2) Oversampling to settle sample imbalance: The original dataset is imbalanced in labels. Hence, I applied Borderline-SMOTE to settle this issue. The code and the result is shown below.

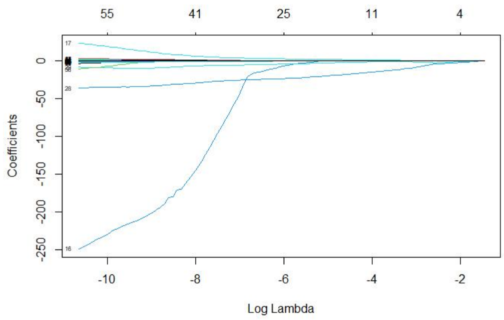

Fig.2 Result of the SMOTE

3. **Transfer Adaboost Model:** Firstly applied Tradaboost to make prediction. And the deep learning network is shown below.

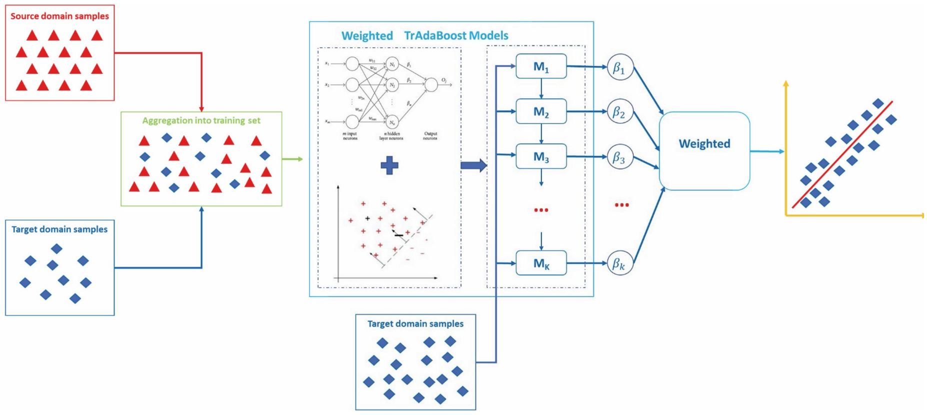

Fig.3 Deep Learning Network of the Project

(1)Initialization:

$$\boldsymbol{w}^1 = (w_1^1, \ldots, w_{n+m}^1), \text{ in which:}$$

$$w_i^1 = \begin{cases} 1/n, \; if\ i = 1, \ldots, n \\ 1/m, \; if\ i = n+1, \ldots, n+m \end{cases}$$

$$\beta = 1/\left(1 + \sqrt{2 \ln n/N}\right)$$

(3) Pseudo Code of the Process:

For $t = 1, \ldots N$

1. Suppose $\boldsymbol{p}^t$ satisfies

$$\boldsymbol{p}^t = \frac{\boldsymbol{w}^t}{\sum_{i=1}^{n+m} w_i^t}$$

2. Call learner, providing it the combined training set T with the distribution $\boldsymbol{p}^t$ over T and the unlabeled data set S. Then, get back a hypothesis $h_t: \boldsymbol{X} \rightarrow \boldsymbol{Y}$ (or [0,1] by confidence)

3. Calculate the error of h_i on T_s:

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t \cdot |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t}$$

4. Set $\beta_t = \epsilon_t/(1-\epsilon_t)$ and $\beta = 1/\left(1 + \sqrt{2 \ln n/N}\right)$.
   Note that, ϵ_t is required to be less than 1/2

5. Update the new weight vector:

$$w_i^{(t+1)} = \begin{cases} w_i^t \beta^{|h_t(x_i) - c(x_i)|}, \; 1 \leq i \leq n \\ w_i^t \beta^{-|h_t(x_i) - c(x_i)|}, \; n+1 \leq i \leq n+m \end{cases}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1, \; \prod_{t=\lceil N/2 \rceil}^{N} \ln(1/\beta_t)\, h_t(x) \geq \frac{1}{2} \prod_{t=\lceil N/2 \rceil}^{N} \ln(1/\beta_t) \\ 0, \; otherwise \end{cases}$$

(4) Result:

Error rate: 0.4876, accuracy is 0.5124.

(5) Analysis:

- The row numbers of source csv are much less than target csv, causing former weights dramatically larger than latter.
- Those source instances that are representative of the target concept tend to have their weights reduced to zero eventually.
- Error rate can not reflect accuracy when the model is an unbalance classifying model.

(6) Further Improvement By **Two-stage Tradaboost**

Update the weight vector into form:

$$w_i^{t+1} = \begin{cases} \dfrac{w_i^t \beta_i^{e_i^t}}{Z_t}, 1 \le i \le n \\ \dfrac{w_i^t}{Z_t}, n+1 \le i \le n+m \end{cases}$$
$$where\ Z_t = \frac{m}{m+n} + \frac{t}{N-1}(1 - \frac{m}{n+m})$$

Using G-mean as criterion:

$$G - mean = \sqrt{\frac{TP^2}{(TP+FN+\epsilon)(TP+FP+\epsilon)}},\ \epsilon \to 0$$

(7) Result: 92.13% of Prediction.